Estimating Time to Image Areas with Steerable 2D Framing Sensors

Garrett Lewellen, Michael Trowbridge, Elly Shao, Christopher Davies and **Russell Knight**
Jet Propulsion Laboratory, California Institute of Technology
{michael.a.trowbridge,elly.j.shao,christopher.j.davies,
russell.l.knight}@jpl.nasa.gov

Abstract

This paper presents a comparison of heuristics used to estimate the amount of time it would take for a spacecraft to image an area using Boustrophedon decomposition (Choset and Pignon 1998). Machine learning techniques are used to characterize algorithmic performance of coverage algorithms. It is shown that an ordinary least-squares linear model is among the most accurate in a set of constant and linear order regression models both in terms of memory consumption and schedule duration. These are demonstrated using the ASPEN planning system (Fukunaga et al. 1997) on the Eagle Eye domain.

Introduction The ASPEN Eagle Eye model is an observational coverage scheduler for space-based, steerable 2D framing instruments. This class includes gimballed telescopes like the ISERV Pathfinder (Stefanov and Evans 2015), proposed Eagle Eye ISS telescope (Knight, Donnellan, and Green 2013) and steerable satellites like ALL-STAR 1/THEIA (Brown et al. 2011). Target polygons are scheduled by rasterizing the polygons into grids of camera-sized instrument tiles at time of observation, then planning a path through instrument tiles (figure 1).

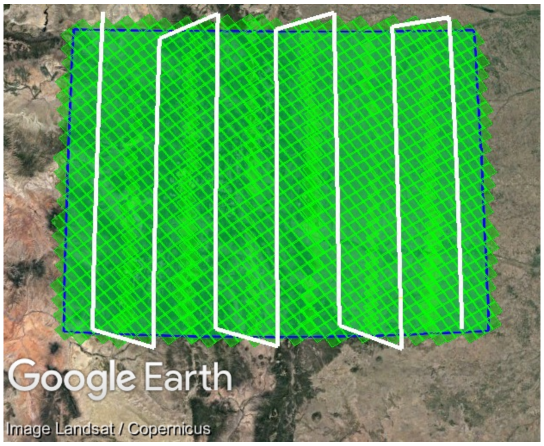

Figure 1: Detailed framing instrument tiling (green) of Colorado (blue), with the Boustrophedon decomposition center-line in white. (Choset and Pignon 1998).

The single area request schedule in figure 1 required 50.3 seconds runtime to compute. The process involves many surface intercepts, polygon operations, constraint checks and function searches. This is reasonable for one request, but optimizing a schedule requires considering multiple possible scheduling times of many requests. Suppose we optimize a meager $n = 10$ area request schedule with an algorithm that has $O\left(n^3\right)$ runtime complexity. At $c_d \approx 50.3$ seconds per detailed scheduling operation, overall schedule optimization would require almost 14 hours runtime.

Abstractly scheduling with a duration heuristic during optimization would make this almost 100 times faster. Instead of computing

Contact author: Michael Trowbridge

the full tiling, pretend that the target polygon is just a single point that the observer stares at for a duration proportional to the target's area (the naive heuristic). This abstract method scheduled figure 1 in $c_a = 5.3$ ms of runtime. The $O\left(n^3\right)$ overall schedule optimization for $n = 10$ area observations takes 5.3 seconds, followed by a final detailed scheduling cost of $c_d n \approx 503$ seconds. More broadly, optimization would be almost 10000 times cheaper; we can afford more of it for a fixed runtime limit.

The problem is that the naive heuristic is too naive. In this case, it underestimated the schedule time needed by 16%. This reserves insufficient resources for the request, so the final schedule must be repaired (likely sub-optimally). The naive estimate could be margined by some constant factor, but an overly pessimistic guess would result in over-reserving of resources and excessive idle time in the final schedule.

Final schedules are expected to have higher quality if the optimizer uses more informed heuristics for activity duration and memory consumption. This paper applies machine learning approaches to build a heuristic model of the Eagle Eye area scheduling algorithms under a variety of observation conditions. A computational experiment will compare the heuristic models against the naive heuristic and measured truth. Varying machine learning models and heuristic feature sets are tested with the goal of finding better alternatives to the naive heuristic.

Related Work

The "Track and Scheduling Problem" is a closely related problem within the area observational coverage planning domain (Lemaître et al. 2002). In this problem, a strip-based Boustrophedon decomposition of the target is planned for an agile Earth-observing satellite. Lemaître et al. showed that optimization of this plan was NP-hard by analogy to the longest path problem, prompting acceptance of sub-optimal solutions and restrictions to the problem. Lin et al. equated strip-based ROCSAT-II observational scheduling to schedule optimization with sequence dependent setup times (Lin et al. 2005), which Pinedo argues is NP-hard by analogy to the Traveling Salesman Problem (Pinedo 2012).

Slew duration between strips is one such sequence-dependent setup time, which Lemaître et. al identified as time dependent and costly to compute (Lemaître et al. 2002). They addressed this with a mean duration heuristic as part of an abstract method of scheduling area observations. They first fix scan direction for an entire overflight, then decompose the target into rectangular strips of length ℓ_i, each representing a sweep of the imager across the target body. Duration d was computed assuming a constant scan rate $\dot{s}$ as

$$d = \sum_{i=0}^{n} \left(\frac{\ell_i}{\dot{s}} + s_{i,i+1} \right) \qquad (1)$$

where $s_{i,i+1}$ is the empirically computed mean duration of an inter-strip transition (turnaround slew) between strips i and $i+1$. This heuristic uses both domain-specific knowledge of the underlying instrument path planning algorithm and a domain-agnostic empirical conditional expectation model of inter-strip transition time.

Our problem is also similar to empirically estimating the computer runtime $G_c(t)$ required by the Iterated Local Search (ILS) approximation algorithm for the Traveling Salesman Problem, where runtime is dependent upon nondeterministic choices and perturbations (Stützle and Hoos 1999).

Our approach borrows techniques from hyperheuristics and machine learning. Cowling et. al introduced the hyperheuristic approach as choosing the best heuristic to estimate the effects of a decision given the current state of the system and cost to evaluate the heuristic (Cowling, Kendall, and Soubeiga 2000). Engelhardt and Chien found online machine learning of planning heuristics in ASPEN to outperform human expert strategies for scheduling the Earth-observing EO-1 spacecraft (Engelhardt and Chien 2000). Fukunaga used genetic algorithms to automate discovery of new boolean satisfyability heuristics in CLASS (Fukunaga 2008). Li et al. augmented a hyperheuristic search with neural networks and logistic regression to quickly classify a solution as worthy or unworthy of evaluating a costly global scoring function (Li, Burke, and Qu 2011).

Formulation

We treat duration d and memory m as response variable outputs of two unknown functions f_d and f_m, which are time, target, observer design and algorithm-varying:

$$d = f_d\left(t, \text{target}, \text{observer}, \text{algorithm}\right) \quad (2)$$

$$m = f_m\left(t, \text{target}, \text{observer}, \text{algorithm}\right) \quad (3)$$

The goal is to cheaply approximate f_d and f_m because they are expensive to compute.

In this study, we constrain the model to one target archetype and spacecraft design. We schedule under varying conditions and compute cheap geometric features $(\theta_{\text{ONA}}, \dot{a}_{\text{IFOV}}, \ldots)$ hoping to discover models where

$$d \approx \hat{d} \ = f_d\left(\theta_{\text{ONA}}, \dot{a}_{\text{IFOV}}, \ldots\right) \quad (4)$$

$$m \approx \hat{m} = f_m\left(\theta_{\text{ONA}}, \dot{a}_{\text{IFOV}}, \ldots\right) \quad (5)$$

The geometric features are explained in the sections that follow.

We also make a simplifying assumption that the results of these models can be scaled linearly to produce sensible predictions for different areas a, repeat frame count F_{Count} and mean observer frame rates F_{Rate} (including tile-to-tile slews) by:

$$\hat{y} = \hat{y}_0 \frac{F_{\text{Count}}}{F_{\text{Rate}}} \quad (6)$$

where $\hat{y}_0$ is the prediction from the model and $\hat{y}$ is the heuristic's prediction of the memory or duration required to satisfy the observation request.

Domain-specific Insight

Tile area changes The target is fixed to the body being observed, but we assume that the observer is on a trajectory such that its position is not fixed. Some overflights are directly over a target, resulting in small, regular tiles. At other times, the observer would have a low, oblique look angle to the target, producing larger, distended tiles. Figure 2 shows how obliqueness affects instrument tile area from a candidate orbit for the proposed Mission to Understand Ice Retreat (MUIR).

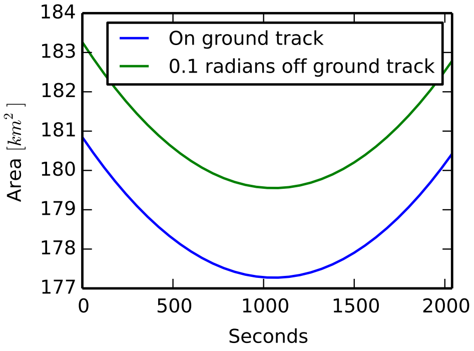

Figure 2: Instrument tile area over the course of an overflight for a tile centered on the ground track and one whose closest approach is 0.1 radians away.

Figure 2 shows that off-nadir angle affects tile area, which affects overall schedule time. We infer that an accurate heuristic model should be queried for each schedule time considered, as off-nadir angle changes over time.

Solar Geometry Ideally, all tiles would be perfectly aligned, leaving no gaps or excessive overlap. In practice, the rotation of the craft, and thus the rotation of the tiles, cannot be chosen freely. One such constraint is the location of the Sun: in order to align the craft's solar panels to the Sun, or avoid pointing instruments at the Sun, the entire craft must rotate. As a result, the tiles will rotate as well.

For example, the ALL-STAR 1 CubeSat's THEIA imagery payload had a rectangular field of view (Brown et al. 2011), lending itself to rectangular grid tiling. Its solar arrays, however, were not articulated - so the craft had a conflicting need to keep the wings aligned to the Sun by rotating about the look vector (+X in figure 3).

Observer Agility Slewing occupies a significant portion of the time required to satisfy an area coverage request. This cost depends on the number of tiles (area ratio), the

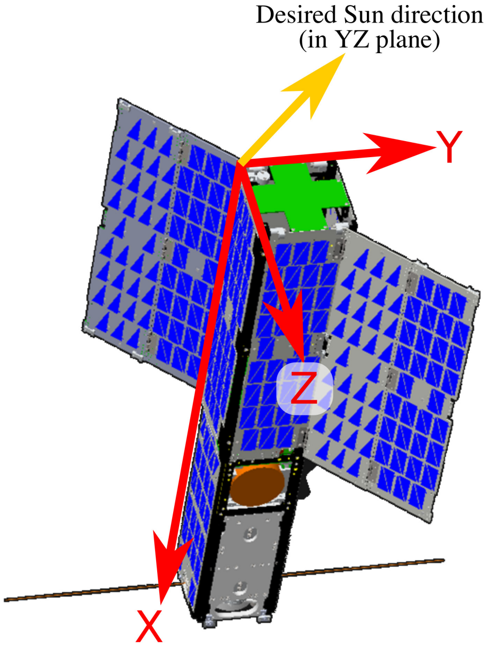

Figure 3: ALL-STAR/THEIA 3U imagery CubeSat (Hayden 2013). Image courtesy Colorado Space Grant Consortium. Used with permission.

path through the tiles (algorithm) and observer agility (slew rates). Treating the area that must be observed to satisfy the target as a great circles polygon on a unit sphere around the observer, we compute the target's area when projected to the observer's unit sphere by following the line integral around the target polygon at the rate specified by the bus agility model.

We will measure the area in seconds because duration is one of the two quantities of interest, and we can use average frame rate to estimate memory from duration. By applying solar orientations at each step along the target polygon bounding curve, we can also address solar rotation costs. This also addresses the fact that different areas of the polygon have different off-nadir angles, and therefore different tile areas on the observed body.

Algorithm Inefficiencies

Turnarounds Continuous scan algorithms that rely on Boustrophedon decompositions require turns between major path segments. The red turnarounds in 4 provide no additional coverage of the target area, and also require rate/acceleration matching at both ends. Small variations can change the amount of schedule time needed:

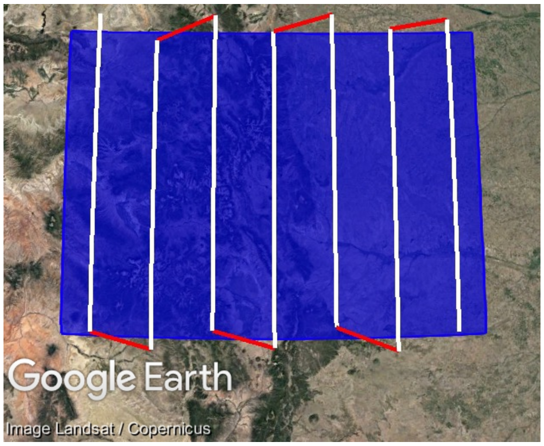

Figure 4: Boustrophedon decomposition (white) with waste turnarounds in red.

- Scan direction across target
- Slower, curved turnarounds
- Varying scan rates per leg

These cost variations cause us to expect that each algorithmic variation will require its own unique heuristic model.

Solar Alignment Policies The grid-based planning algorithm tested in this paper defines its grid at the start of a window under consideration. Spacecraft orientation, however, is defined by schedule-time basis vectors. The primary vector in figure 3 is $\hat{\mathbf{x}}$ toward the grid point, with the secondary vector $\hat{\mathbf{s}}_{\text{des}} = R_1\left(-45°\right)\hat{\mathbf{y}}$ as the projection of the Sun vector in the YZ plane. Solar alignment is part of the slew cost, so this algorithm is expected to be sensitive to solar geometry. Solar geometry features that may inform the model:

- Solar Elevation Angle (SEA)

- Observer-target-Sun phase angles

In the worst case, the conflict between continuous solar alignment and planning grid alignment causes the instrument field of view to rotate significantly out of alignment with the planning grid, resulting in gaps in coverage that require a second pass.

Potential Features

In order to estimate different outputs of the scheduling process, a handful of different features were considered which were believed to influence the response variables we wish to model. These features can be roughly categorized as point, perimeter, or area-based:

Point-based These features require only a measurement at a single point.These measurements are quick, but provide little information about the area as a whole.

- Off nadir angle
- Solar off-zenith angle at target
- Cross-track angle
- Along-track angle

Perimeter-based These features require measurements along the entire perimeter P of all target polygons in the observation request. The combine observer agility (slew rate) with observational geometry (skew, rotation).

- Slew time line integral $\oint_P dt_{\text{slew}}$
- IFOV tile area line integral $\oint_P da_{\text{IFOV}}$

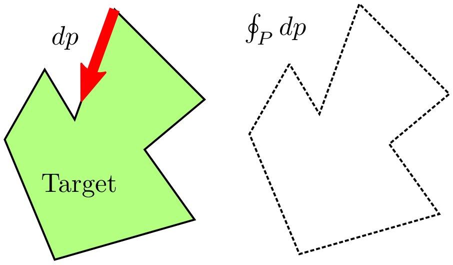

Figure 5: Perimeter integral

Table 1: Feature complexity (p points)

Feature	Complexity
Off nadir angle	$O(1)$
Solar off-zenith angle	$O(1)$
Cross-track angle	$O(1)$
Along-track angle	$O(1)$
$\oint_P da_{\text{IFOV}}$	$O(\lvert p\rvert)$
$\oint_P dt_{\text{slew}}$	$O(\lvert p\rvert)$
Polygon area a	$O(\lvert p\rvert)$
$\dot{a}_{\text{IFOV}}$	$O(1)$

Area-based These features require measurements of the entire interior. These measurements are slow, but provide the most information about the area.

- Polygon area a (km^2)
- IFOV Tile area derivative $\dot{a}_{\text{IFOV}}$

Parameters relating to time to acquire a tile were ignored as typically negligible compared to slew costs. Table 1 summarizes complexity of evaluating these features for a polygon of p points.

Models

This work focused on constant and linear models given their ease of implementation and desirable memory and runtime properties over non-linear models. Individual models were evaluated in Excel or in Python using scikit-learn. The following is a brief overview of the models considered:

Naive Heuristic Assumes that the target area is evenly divided into instrument tiles, and that scheduling time d and memory consumption m depend only on the number of data takes n_{tiles}.

$$n_{\text{tiles}} = \frac{a_{\text{target}}}{a_{\text{IFOV}}\left(t_{\text{center}}\right)} \frac{c_{\text{images}}}{\text{tile}} \tag{7}$$

This implies an assumption that all instrument tiles have the same area a_{IFOV}, the area of a tile at the target bounding box center at time t_{center} of the target access/scheduling window being considered.

The estimated memory consumption $m\left(t_{\text{center}}\right)$ is

$$m\left(t_{\text{center}}\right) = c_1 n_{\text{tiles}} \frac{m_{\text{bits}}}{\text{image}} \tag{8}$$

and the estimated duration $d\left(t_{\text{center}}\right)$ is

$$d\left(t_{\text{center}}\right) = c_2 n_{\text{tiles}} \frac{\Delta t}{\text{image}} \tag{9}$$

Shortfalls of the Naive Heuristic:

- Ignores algorithm-specific inefficiencies
- Ignores changes in tile area within an access
- Assumes shortest path between tiles
- Hand-tuned parameters c_1, c_2

Ordinary Least Squares Takes a linear combination of features x and weights θ and minimizes the residual squared error with the observed response y:

$$\underset{\theta}{\operatorname{argmin}} \sum_i \left(\langle x_i, \theta\rangle + \beta - y_i\right)^2 \tag{10}$$

Different regularization strategies did not produce materially different results compared to Ordinary Least Squares, and hence we omit discussion of Ridge, Lasso, and Elastic Net methods.

Support Vector Regression The traditional support vector machine framework for classification can be retooled to support regression, which creates a quadratic program that needs to be solved. Cf. (Smola and Vapnik 1997). Two kernels are evaluated in this paper - linear and radial basis functions (RBF).

Empirical Conditional Expectation This approach reports the sample mean of a response variable based on an empirical conditional probability table constructed from binned values of the conditioners from the training set $\mathcal{D}$.

$$\mathbb{E}(Y|\theta, \phi) = \sum_{y \in \mathcal{D}} y \hat{p}(y|\theta, \phi) \tag{11}$$

If one dimension of the query point is not in the CPT, then the conditional expectation on the available dimension is used, and if both dimensions are unavailable, then an unconditioned expectation is used.

Generalized Regression Neural Networks Computes weighted average for the response using a Gaussian kernel estimate of the joint probability over training set $\mathcal{D}$. Cf. (Specht 1991).

$$f(x|X, Y) = \frac{\sum_i^n y_i \exp\left(-\langle x - x_i\rangle / 2\sigma^2\right)}{\sum_i^n \exp\left(-\langle x - x_i\rangle / 2\sigma^2\right)} \tag{12}$$

Smoothing parameter σ determined by exhaustive grid search.

Methodology

The training and test data was collected by generating 2000 polygons of random area $a = U\left(10, 100000\right)$ km^2 at random center points on Earth (figure 6), then scheduling each one independently. Scheduling was constrained to one overflight. Target polygons that were not 100% satisfied were considered outliers and removed from the data set.

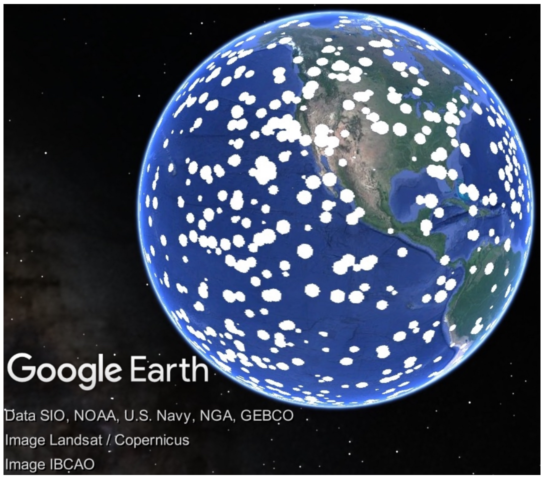
Google Earth
Data SIO, NOAA, U.S. Navy, NGA, GEBCO
Image Landsat / Copernicus
Image IBCAO

Figure 6: Target polygons sampled

Response variable d (duration) was measured by subtracting the start time of the first tile from the end time of the final tile. Response variable m (memory) is the sum of all tile memory reservations. Features are computed at the center time of the access window. Features and response variables are printed to console out by the ASPEN user function `gatherAreaEstimatorData`, then assembled into a comma-separated-values file by the python ASPEN run management script `python-driver/src/main.py`.

Models are fit to the data and evaluated in Python. The Least Squares and SVR families used scikit-learn implementations. Custom Python implementations were used for ECE and GRNN. The Naive Heuristic was computed in ASPEN.

Reasonability of each potential feature was examined by computing correlation coefficients with duration and memory. Features that were reasonably correlated with the response variables were hand-selected as two-parameter features for the models.

Models are evaluated by random subsampling (Han and Kamber 2006), with an 80% training/20% test split for each of the $n = 60$ iterations. Accuracy of the models is evaluated by mean error μ and root mean squared error (RMSE) σ against the test sets. A good heuristic model based on sufficient data is expected to have zero mean error and a small RMSE against the test set.

Results

Feature Identification

Four viable model features (one point feature, two perimeter features, and one area feature) were identified (table 2):

- Off-nadir angle
- $\oint_P da_{\mathrm{IFOV}}$: perimeter area line integral
- $\oint_P dt_{\mathrm{slew}}$: perimeter slew time line integral
- $\dot{a}_{\mathrm{IFOV}}$

Selected features are discussed in this section.

Table 2: Coefficients of Determination

Feature	Duration	Memory
Area line integral	0.89	0.92
Slew line integral	0.78	0.81
Off-nadir angle	0.43	0.44
Area	0.32	0.33
$\dot{a}_{\mathrm{IFOV}}$	0.12	0.12
Along-track angle	0.08	0.07
Solar emission angle	0.07	0.07
Cross-track angle	0.00	0.00

Naive Heuristic Figure 7 shows weak correlation between the Naive Heuristic and true duration. Area of the target is important, but clearly not the only important driver. A simple tile to target area ratio is insufficient to predict schedule time/memory requirements.

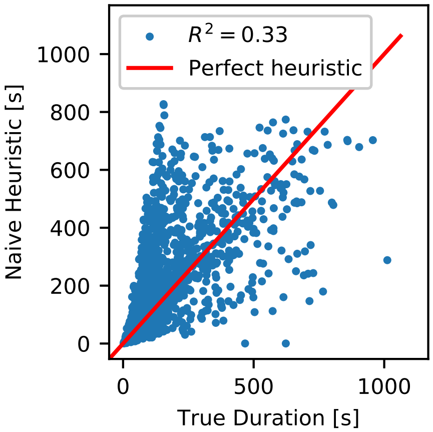

Figure 7: Correlation between the Naive Heuristic and the true duration (as-scheduled)

Solar Off-Zenith Angle Solar Off-Zenith angle and memory (figure 8) have two correlation regions. They are weakly correlated for low to moderate Solar off-zenith angles, but uncorrelated at high Solar off-zenith angles. Solar off-zenith angle may be more useful if managed by a hyperheuristic.

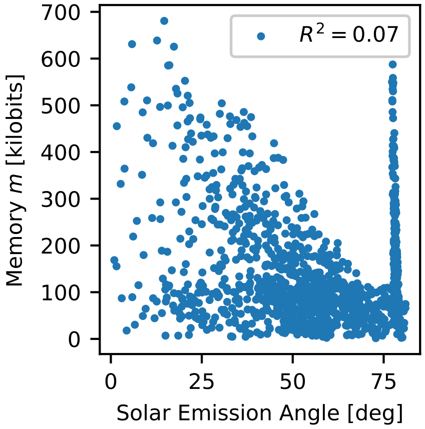

Figure 8: Solar off-zenith angle vs. memory

Perimeter integrals The instrument tile area perimeter integral (figure 9) and the slew time perimeter integral (figure 10) were the most correlated features discovered in this study. Unsurprisingly, they are both present in the best performing feature sets in table 5.

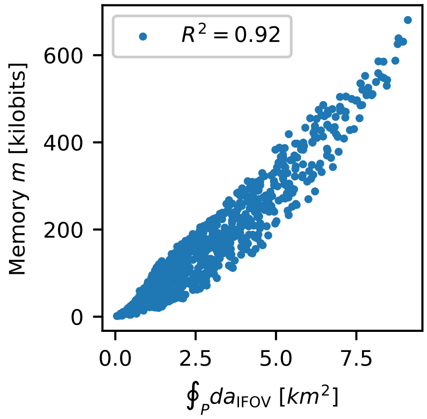

Figure 9: $\oint_P da_{\text{IFOV}}$ vs. memory

The slew time line integral resembles $y = x^2$. Future work should examine logarithmic features of the form $y = \log\left(\oint_P dt_{\text{slew}}\right)$.

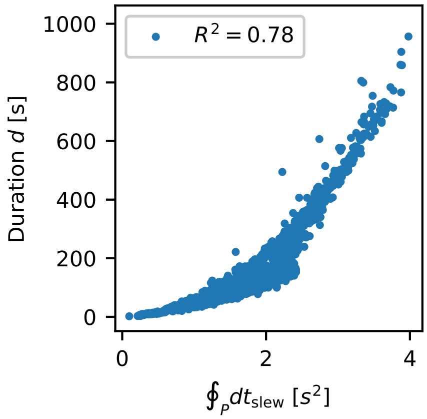

Figure 10: $\oint_P dt_{\text{slew}}$ vs. duration

Duration d

Table 3 shows that all of the models are significantly more accurate than the Naive Heuristic. The ECE model has the worst test set error of the empirical models evalutated.

Table 3: Mean error (μ) and RMSE (σ) of models in estimating duration, features: $\dot{a}_{\text{IFOV}}$, $\oint_P da_{\text{IFOV}}$.

	Train		Test	
Model	μ	σ	μ	σ
OLS	0.00	55.61	-0.65	54.88
Linear SVR	-3.82	55.62	-3.66	57.60
RBF SVR	-2.23	52.83	-3.26	52.59
ECE	0.00	37.48	4.24	81.23
GRNN	1.44	48.00	1.31	52.56
Naive	0.00	0.00	51.81	173.02

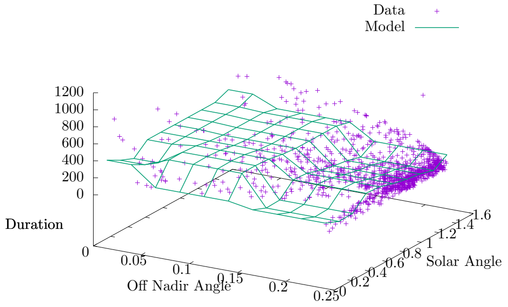

Figure 11: Off nadir angle and solar angle raw data obtained from running Eagle Eye (points), and the corresponding empirical conditional expectation model fit (lines).

Inspection of the Lasso regression coefficients revealed that the off nadir angle and solar angle were prominent features. Inspection of the ECE output suggests that there may be an underlying smooth surface that might be captured by a nonlinear model.

Table 5: Mean error (μ) and RMSE (σ) of OLS models by feature set (test only).

	Duration [s]		Memory [kbits]	
Feature Set	μ	σ	μ	σ
$\dot{a}_{\mathrm{IFOV}}$, $\oint_P da_{\mathrm{IFOV}}$	-0.01	54.94	0.23	34.28
$\dot{a}_{\mathrm{IFOV}}$, $\oint_P da_{\mathrm{IFOV}}$, $\oint_P dt_{\mathrm{slew}}$	0.52	45.65	0.08	24.84
Off nadir angle, $\oint_P dt_{\mathrm{slew}}$	0.07	61.44	-0.05	40.02
Off nadir angle, Solar angle, $\oint_P dt_{\mathrm{slew}}$	0.23	61.64	-0.32	40.11
$\dot{a}_{\mathrm{IFOV}}$, $\oint_P dt_{\mathrm{slew}}$	-0.17	70.81	0.12	49.65
Off nadir angle, Solar angle	-0.89	126.55	-1.07	92.42
Cross-track angle, Along-track angle	1.93	163.59	-1.47	117.80

Memory Consumption m

Memory consumption models follow the same basic trends as the duration models. All models are significantly better than the Naive Heuristic.

Table 4: Mean error (μ) and RMSE (σ) of models in estimating memory, features: $\dot{a}_{\mathrm{IFOV}}$, $\oint_P da_{\mathrm{IFOV}}$.

	Train		Test	
Model	μ	σ	μ	σ
OLS	0.00	34.08	-0.19	33.62
Linear SVR	-4.05	34.57	-3.70	34.68
RBF SVR	-2.59	31.18	-2.10	32.28
ECE	-0.00	22.00	2.85	52.31
GRNN	1.07	28.01	0.98	31.97
Naive	0.00	0.00	-17.28	104.57

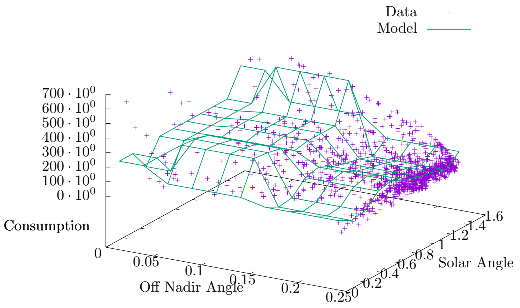

Figure 12: Off nadir angle and solar angle raw data obtained from running Eagle Eye (points), and the corresponding empirical conditional expectation model fit (lines).

Comparison of Feature Sets

Table 5 shows the impact of changing the features that the model is based on. In each feature set variation, ordinary least squares (linear regression) was used as the model. The best estimates came from feature sets including a perimeter integral feature and the instrument tile area derivative $\dot{a}_{\mathrm{IFOV}}$.

Discussion

Model performance varied significantly depending based on the feature set selected. The feature set $\left(\dot{a}_{\mathrm{IFOV}}, \oint_P da_{\mathrm{IFOV}}\right)$, for example, is dominated by an approximately linear correlation of $\oint_P da_{\mathrm{IFOV}}$ with memory and duration (figure 9). Nonlinear regression techniques produce approximately linear models for this feature set, all of which are comparable with ordinary least squares.

The feature set (Off nadir angle, $\oint_P dt_{\mathrm{slew}}$), however, has nonlinear correlations with duration and memory. Radial basis function SVR and GRNN were significantly more accurate than ordinary least squares with that feature set.

While radial basis function SVR and GRNN had the best performance with varied feature sets, they each have drawbacks. The GRNN has high computational complexity and would likely require GPU implementation in practice. Limiting scope to $\pm 4\sigma$ around the independent variable tuple made the GRNN faster, but increased the RMSE by approximately 50%. Radial basis function SVR is slightly slower than linear kernel SVR or ordinary least squares and more complicated to implement.

Challenges in sample selection

The random polygon of random area method of selecting sample targets (figure 6) was the third method attempted in this study. Binning the Earth by latitude/longitude made area and off-nadir angle unrealistically latitude-dependent (figure 13).

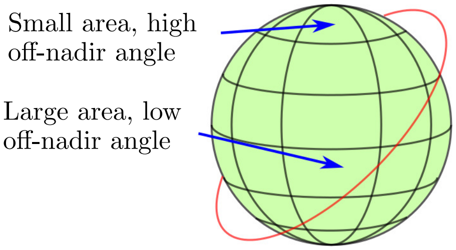
Small area, high off-nadir angle
Large area, low off-nadir angle

Figure 13: Unintended latitude dependency

Icosahedron discretization had no latitude dependency, but it restricted the feature space sampling to a small manifold because the icosahedron faces had similar area.

Future Work

The primary focus of future work is on ensuring uniform sampling of the off nadir and solar angle space, as the current data set is biased toward high off-nadir angles, leaving some parts of the model more accurate than others. Second to that is shortening the data acquisition period and ensuring that the process can be interrupted, resumed, and distributed across multiple computing resources. Finally, the start and end points of the tour can be estimated to improve planning across multiple areas.

It may also be possible to improve the instrument tile area rate of change feature $\dot{a}_{\text{IFOV}}$. The correlation plots are symmetric about the line $\dot{a}_{\text{IFOV}} = 0$, and each side appears to have an upper bound of $d = c/\dot{a}_{\text{IFOV}}$. The feature $\log|\dot{a}_{\text{IFOV}}|$ may be more linearly correlated with duration and memory than $\dot{a}_{\text{IFOV}}$ is.

This study should also be repeated with a more rigorous hyperheuristics structure. Exposing all geometric functions that can affect visibility and relative dynamics between the observer and target may allow for automated feature selection, model selection and hyperparameter configuration.

Conclusions

A handful of constant and linear order regression models were evaluated against a baseline method to predict the duration and consumption of a scheduling call under different conditions. It was found that both schedule duration and memory consumption could be improved by a factor of 3 by using an ordinary least squares linear model. Future work will focus on improving these metrics, investigating start/stop position prediction and automating feature and model selection.

Acknowledgements

The research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

Special thanks to Brian Sanders of the Colorado Space Grant Consortium for authorizing the use of ALL-STAR images and data in this paper.

The authors would like to thank the reviewers for their constructive feedback, which was quite helpful in improving this paper.

References

Brown, J.; Pack, R.; Murphy, T.; and Kophs, R. 2011. All-star critical design review: Systems and management.

Choset, H., and Pignon, P. 1998. Coverage path planning: The boustrophedon cellular decomposition. In *Field and service robotics*, 203–209. Springer.

Cowling, P.; Kendall, G.; and Soubeiga, E. 2000. A hyperheuristic approach to scheduling a sales summit. In *International Conference on the Practice and Theory of Automated Timetabling*, 176–190. Springer.

Engelhardt, B., and Chien, S. 2000. Empirical evaluation of local search methods for adapting planning policies in a stochastic environment. In *Workshop on Local Search for Planning and Scheduling*, 108–119. Springer.

Fukunaga, A.; Rabideau, G.; Chien, S.; and Yan, D. 1997. Aspen: A framework for automated planning and scheduling of spacecraft control and operations. In *Proc. International Symposium on AI, Robotics and Automation in Space*.

Fukunaga, A. S. 2008. Automated discovery of local search heuristics for satisfiability testing. *Evolutionary computation* 16(1):31–61.

Han, J., and Kamber, M. 2006. *Data Mining: Concepts and Techniques*. The Morgan Kaufmann series in data management systems. Elsevier.

Hayden, I. 2013. All-star structure overview.

Knight, R.; Donnellan, A.; and Green, J. J. 2013. Mission design evaluation using automated planning for high resolution imaging of dynamic surface processes from the iss. In *Proceedings of the 8th International Workshop of Planning and Scheduling for Space (IWPSS-2013)*. Jet Propulsion Laboratory, National Aeronautics and Space Administration.

Lemaître, M.; Verfaillie, G.; Jouhaud, F.; Lachiver, J.-M.; and Bataille, N. 2002. Selecting and scheduling observations of agile satellites. *Aerospace Science and Technology* 6:367–381.

Li, J.; Burke, E. K.; and Qu, R. 2011. Integrating neural networks and logistic regression to underpin hyper-heuristic search. *Knowledge-Based Systems* 24(2):322–330.

Lin, W.-C.; Liao, D.-Y.; Liu, C.-Y.; and Lee, Y.-Y. 2005. Daily imaging scheduling of an earth observation satellite. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* 35(2):213–223.

Pinedo, M. 2012. *Scheduling: Theory, Algorithms, and Systems*. Springer New York.

Smola, A., and Vapnik, V. 1997. Support vector regression machines. *Advances in neural information processing systems* 9:155–161.

Specht, D. F. 1991. A general regression neural network. *IEEE transactions on neural networks* 2(6):568–576.

Stefanov, W., and Evans, C. 2015. Data collection for disaster response from the international space station. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 40(7):851.

Stützle, T., and Hoos, H. H. 1999. Analyzing the run-time behaviour of iterated local search for the tsp. In *III Metaheuristics International Conference*.